

## Parsing

Phase	Input	Output
Lexer	string of characters	string of tokens
Parser	string of tokens	Parse tree

### Parser Roles

- (Parse tree) 3 ways (string of tokens) in 1
- (Valid and non valid string of tokens) in 2 - 2

### CFG (Context Free grammars)

\* Programming language constructs have recursive structure.

\* Context-free grammars are natural notation for describing this recursive structure

### CFG consist of

1) set of terminals (T)

2) set of non-terminals (NT)

3) set of productions (P)

3) A start symbol  $S$  (a non-terminal)

4) set of production

مع دلالة: تعيد (replace) في ال (string) الناتج.

$S \rightarrow Ab$

Terminals  $a, b, c$

$A \rightarrow Bc$

Non-terminals  $a, S, A, B, C$

$B \rightarrow Cc$

$C \rightarrow \epsilon$

↳ that string doesn't accept except

$b's$  or  $C's$ .

مع دلالة ال Grammar

1- يعرف كل ال (expressions) التي تولد اللغة الخاصة بـ.

2- يعمل إستعداد أو (مستقبل لكل ال (expressions) التي مشرعية.

How to Derive any Expression

1) ابدأ بـ (string) يحتوي الرمز " $S$ " (start symbol)

2) استبدل ال Non-terminal وفتح ما بين قوسين Terminal or  $\epsilon$

3) كرر رقم (2) حتى تصل لعالة عدم وجود Non terminal في ال (string)

\* مقدرش آید (replacement) لا (terminal) بواسطه ال (Production tool).

\* ولا لا يوجد (terminal) ولا لا يوجد.

\* if you wanna to make derivation for  
if id then id else id  $P_i$

$Expr \rightarrow$  if  $Expr$  then  $Expr$  else  $Expr$   $P_i$

$\rightarrow$  if (if then  $Expr$  else  $Expr$   $P_i$ )

then  $Expr$  else  $Expr$   $P_i$

\* which of strings are in language of given CFG

1)  $abcba \Rightarrow \text{xx}$

2)  $acca \Rightarrow \text{xx}$

3)  $aba \Rightarrow \text{xx}$

4)  $abcbcb \Rightarrow \text{xx}$

$S \rightarrow aXa$

$X \rightarrow \epsilon$

$| bY$

$Y \rightarrow \epsilon$

$| cXc$

3

هم صياغة منطقية

a)  $abcba$

لتحديد ما إذا كانت (CFG) التي أعدها عالمي في الصفحة السابقة تقبل مع المتابعة  $a$  أم لا.



→  $abccca$  (نقاط)  
unaccepted

b)  $acca$



لا تقبل أيضا

c)  $aba$



$aba \Rightarrow$  accepted

~~دالة~~  
d)



$abcbccca$

not accepted





ambiguity (متعدد المعاني) في ال (sheets).

\* لا يوجد خطرات خاصہ کی کتب وجود ال (Ambiguous)

بجانب العمل الآخر، إنك تكتب (rewrite) له (Grammar) بحيث هي (Avoid Ambiguity).



دی (Ambiguous) ہے  $A \rightarrow AaA \mid AbA$

$$A \rightarrow C \mid S$$

العلل سه نمکبدی  $A \times A$  و  $B$  در یکدیگر

$$S \rightarrow A \cup A \setminus B$$
$$B \otimes A \otimes B \rightarrow A \otimes A$$
$$A \rightarrow c/s$$

~~MISS~~ The dangling else

else matches the closest then.

$S \rightarrow MIF$  (all then matched with an else)

10 IF (since there is unmatched)

$$\text{MIF} \Rightarrow \text{IF } E \text{ then } \frac{\text{MIF}}{\text{if-then else}} \text{ else } \frac{\text{MIF}}{\text{if-then else}}$$


\* if E then E else E

→ if E then UIF else MIF

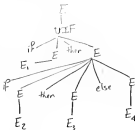
if then

لا يقبل (if) ولا (else) لا يقبل

→ if E then MIF else UIF

if then else

accepted



[I]

منه من أجل وثائق الـ (Compiler) إنه يعمل (handling)  
لا (non-valid program or errors)

### Panic mode [1]

منه يقدم عمل تتاحل لا (tokens) من أجل لتفهم  
والجمله تتحلل العجله بسيطه.

[Ex]

$$(1 + + 2) + 3$$

منه من المثال للبين 1 ثم + ثم تتاحل + القارمه  
حتى تتحلل لـ 2 ويكمل العجله ويظهر رسالة ان فيه  
(syntax error) في مكان معين.

$$1 + \text{حتى تتحلل} 2$$

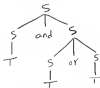
— ينتج (error production)  
Lecture 5 (slides) last

3 - pages.



# p. \* precedence

→ ! به الى إمتنع الأول



→ لها أولوية أعلى (or) لها أولوية أعلى.



له (and) لها أولوية أعلى

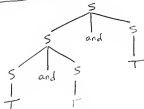
→ لو عايز أجبره في فعل (and) لها أولوية أعلى  
(or) عندك

$$S \rightarrow S \text{ or } S \mid L$$

$$L \rightarrow \text{true and } L \mid \text{true}$$



\* Left Associative



\* Right Associative



(10)